International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 1 | Jan-2017 www.irjet.net p-ISSN: 2395-0072

Simulation based design and analysis of combined effect of various data security techniques used during the transmission of 128-bit digital data generated from 128-bit data generation unit written in VHDL Code using Xilinx ISE 9.2i software

[1]Paresh Kumar Pasayat, Asst.Professor, IGIT Government Engineering College, Odisha, India
[2]Sony Naik, M.Tech student, IGIT Government Engineering College, Odisha, India

---

Abstract - *In the recent years, data security has become an important issue due to the hacking of data. In order to overcome this problem, a technique known as encryption has come up as a solution, and plays an important role in information security system. The desired 128-bit data is encrypted using modified DES and Hamming (224,128) code technique in addition to the use of modified iterated product cipher to produce 256-bit encrypted data. As the proposed design is having the combined effect of modified DES, Hamming (224,128) code and modified iterated product cipher data security techniques, the security level is very high as compared to the design having individual data security technique. Due to the increment of key size from 56-bits to 112-bits in modified DES, the design is more resistive to the Brute-Force Attack. The other advantages of the proposed work are: Confidentiality, Authentication and Integrity. This can be used in the field of Automated teller machine (ATM) transactions, Banking sector, Military sector and protecting confidential company information. The proposed work is done by using VHDL language. The code is tested and simulated using Xilinx ISE9.2i software.*

Key Words: **ALU (Arithmetic Logic Unit), Encryption, Decryption, VHDL (Very High speed Integrated Circuit Hardware Description Language).**

1. INTRODUCTION

For the transmission of 128-bit digital data, the first thing is to generate the data using data generation unit. The data generation unit consists of three components. First component is the control unit which generates the control signals. Second component is the data path unit which consists of one Arithmetic and Logic Unit (ALU) and backup unit and the ALU performs different operations based on the value of the control signal. Third component is the memory unit which is used to store the 128-bit data. The memory unit is controlled by one chip enable signal (C). When the value of C is '1', then the memory unit gives output. It gives no output when the value of C is '0'. When C='0', the result is obtained from the backup unit. If the digital data is transmitted directly without using encryption technique, then there is more probability of hacking and corruption of data by the attacker. Due to which, the various data security techniques have been designed by the designer to provide security to the data. The transformation of original data into a data which is not in the readable form is known as encryption and the process of reversing it back to a readable form is known as decryption. The proposed design shows how the 128-bit data is transmitted into space after doing encryption using modified DES and Hamming (224,128) code techniques.

1.1 Project Model

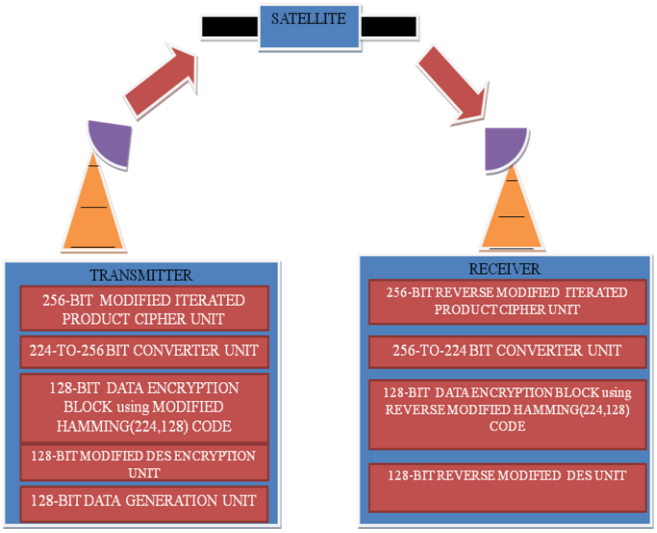

Fig-1: Project Model of the proposed work

1.2 LOGIC USED IN THE PROPOSED DESIGN

The flow chart of the proposed design is given as follows:

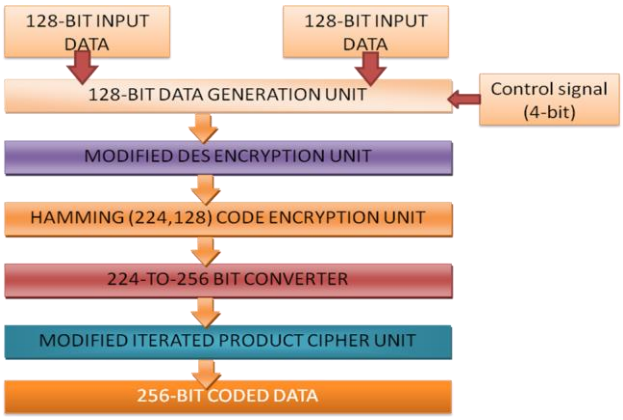

Fig-2: Flow chart of the proposed design

The logic used in the modified DES encryption is given as follows:

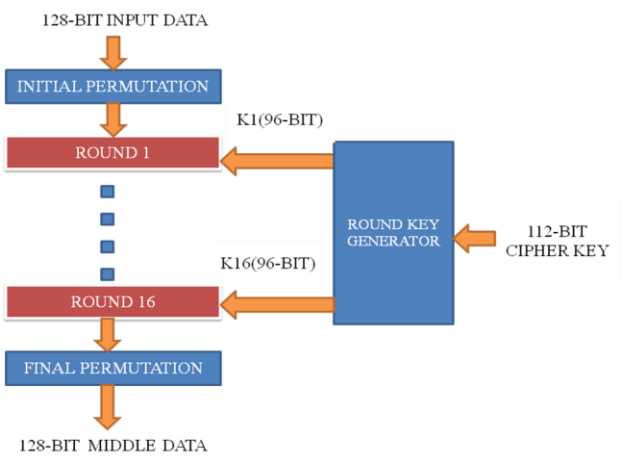

Fig-3: Flow chart of modified DES encryption

ROUND KEY GENERATOR:

```
KEY_OUT_1 <= KEY_IN(0) & KEY_IN(95 DOWNTO 1);
KEY_OUT_2 <= KEY_IN(1)& KEY_IN(0) & KEY_IN(95
             DOWNTO 2);
KEY_OUT_3 <= KEY_IN(2) & KEY_IN(1)& KEY_IN(0) &
             KEY_IN(95 DOWNTO 3);
KEY_OUT_4 <= KEY_IN(3) & KEY_IN(2) & KEY_IN(1)&
             KEY_IN(0) & KEY_IN(95 DOWNTO 4);
KEY_OUT_5 <= NOT KEY_IN(95 DOWNTO 0);
KEY_OUT_6 <= NOT KEY_IN(95 DOWNTO 0);
KEY_OUT_7 <= NOT KEY_IN(95 DOWNTO 0);
KEY_OUT_8 <= KEY_IN(45) & KEY_IN(95 DOWNTO 1);
KEY_OUT_9 <= KEY_IN(48) & KEY_IN(95 DOWNTO 1);
KEY_OUT_10 <= KEY_IN(41) & KEY_IN(95 DOWNTO 1);
KEY_OUT_11 <= KEY_IN(45) & KEY_IN(94 DOWNTO 1) &
              KEY_IN(90);
KEY_OUT_12 <= KEY_IN(91) & KEY_IN(95 DOWNTO 1);
KEY_OUT_13 <= KEY_IN(45) & KEY_IN(95 DOWNTO 1);
KEY_OUT_14 <= KEY_IN(46) & KEY_IN(95 DOWNTO 1);
KEY_OUT_15 <= KEY_IN(40) & KEY_IN(95 DOWNTO 1);
KEY_OUT_16 <= KEY_IN(1) & KEY_IN(95 DOWNTO 1);
```

Here KEY_IN is the 112-bit cipher key and KEY_OUT is the 16 nos. of keys generated from the Round Key Generator.

The logic used for the implementation of the different blocks of modified DES is given as follows:

INITIAL PERMUTATION UNIT:

```
IPU_DATA_OUT(0) <= IPU_DATA_IN(127);
IPU_DATA_OUT(1) <= IPU_DATA_IN(126);
IPU_DATA_OUT(2) <= IPU_DATA_IN(125);
IPU_DATA_OUT(3) <= IPU_DATA_IN(124);
IPU_DATA_OUT(123 DOWNTO 4) <= IPU_DATA_IN(123
                              DOWNTO 4);
IPU_DATA_OUT(124) <= IPU_DATA_IN(3);
IPU_DATA_OUT(125) <= IPU_DATA_IN(2);
IPU_DATA_OUT(126) <= IPU_DATA_IN(1);
IPU_DATA_OUT(127) <= IPU_DATA_IN(0);
```

Here IPU_DATA_IN and IPU_DATA_OUT are the 128-bit input and output datas of the initial permutation block.

16 ROUNDS IN DES:

DES uses 16 rounds of operations. Each round consists of following units performing different types operations.

BIT SEPERATOR UNIT:

The output of the initial permutation block is given to the bit separator unit.

```
BS_OUT_DATA_ONE<=BS_IN_DATA(127 DOWNTO 64);
BS_OUT_DATA_TWO<=BS_IN_DATA(63 DOWNTO 0);
```

FIESTAL CIPHER UNITXOR UNIT:

The block diagram showing the Fiestal Cipher Unit is given as follows:

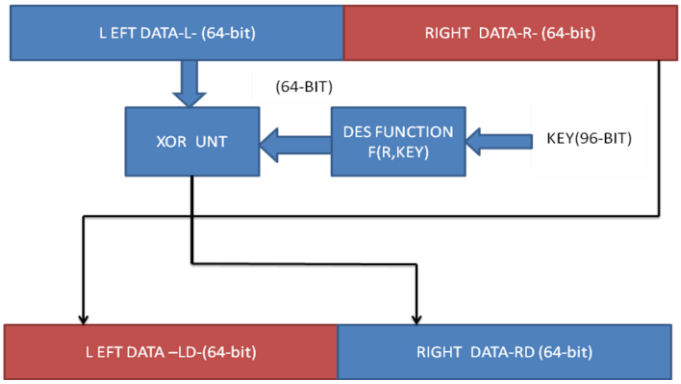

Fig-4: Flow chart of Fiestal Cipher Unit

Then, the two outputs of the bit separator unit is given to the Fiestal Cipher Unit and the different performed are given as follows.

```
S_EX_P_BOX_OUTPUT := FCU_IN_DATA( 31 DOWNTO 0) &
FCU_IN_DATA(63 DOWNTO 32) &
"00000000000000000000000000000000";
```

```
S_XOR_OUTPUT :=  S_EX_P_BOX_OUTPUT XOR KEY_INPUT;

S_SUBT_BOX_OUTPUT :=  S_XOR_OUTPUT(31 DOWNTO 0)
& S_XOR_OUTPUT(63 DOWNTO 32) & S_XOR_OUTPUT(95
DOWNTO 64);

FCU_OUT_DATA <= S_SUBT_BOX_OUTPUT(15 DOWNTO 0)
& S_SUBT_BOX_OUTPUT(31 DOWNTO 16) &
S_SUBT_BOX_OUTPUT(63 DOWNTO 32);
```

The different operations that are performed in the Fiestal Cipher Unit are XOR operation, swapping operation, bit append operation.

DES FUNCTION[F(R,KEY)]:
The different units of the DES function used in the Fiestal Cipher is given as follows:

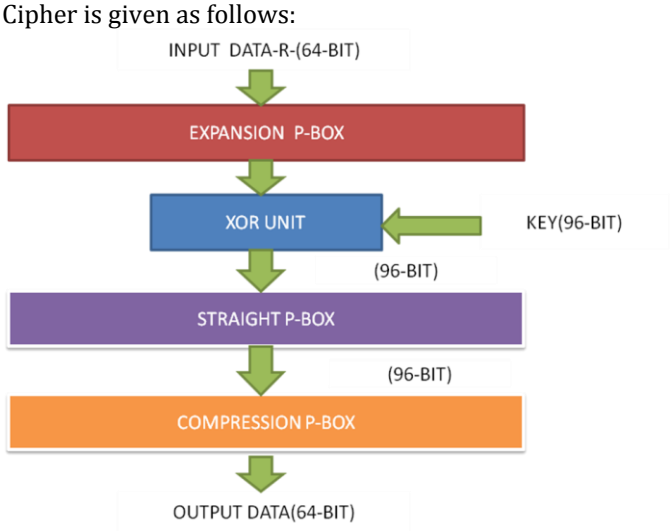

Fig-5: Flow chart of DES function used in the Fiestal Cipher

XOR UNIT:

```
XU_OUT_DATA <= XU_IN_DATA_ONE XOR
                XU_IN_DATA_TWO;
```

SWAP UNIT:

```
SU_OUT_DATA_ONE   <= SU_IN_DATA_TWO;
SU_OUT_DATA_TWO <= SU_IN_DATA_ONE;
```

BIT APPEND UNIT:

```
BAU_OUT_DATA <= BAU_IN_DATA_ONE &
                BAU_IN_DATA_TWO;
```

There are 16 nos. of round in the modified DES and after the completion of the round 16, the final permutation operation is performed.

FINAL PERMUTATION:

```
FPU_DATA_OUT(0) <= FPU_DATA_IN(127);
FPU_DATA_OUT(1) <= FPU_DATA_IN(126);
FPU_DATA_OUT(2) <= FPU_DATA_IN(125);
FPU_DATA_OUT(3) <= FPU_DATA_IN(124);
FPU_DATA_OUT(123 DOWNTO 4) <= FPU_DATA_IN(123
                              DOWNTO 4);
FPU_DATA_OUT(124) <= FPU_DATA_IN(3);
FPU_DATA_OUT(125) <= FPU_DATA_IN(2);
FPU_DATA_OUT(126) <= FPU_DATA_IN(1);
FPU_DATA_OUT(127) <= FPU_DATA_IN(0);
```

Algorithm For Hamming (224,128) code Encryption Unit
Step 1
First, 128-bit data is divided into 32 nos. of words each consisting of 4-bit data.
Step 2
The 7-bit Hamming (7,4) code encoding technique is applied to each word. For each word, the encoding unit generates 7-bit encoded data. The logic for implementing the Hamming code technique is given as follows:
Suppose, the 4-bit data (B) to be encoded is B3B2B1B0 and the 7-bit Hamming code (H) generated is H6H5H4H3H2H1H0.
Here, the value for each bit of H is given as follows:
H6 = B3 xor B2 xor B0
H5 = B3 xor B1 xor B0
H4 = B2 xor B1 xor B0
H3 = B3
H2 = B2
H1 = B1
H0 = B0
Step 3
After that the Hamming codes corresponding to each word are appended to form the desired 224-bit encoded data.

The block diagram showing the encryption process using the above algorithm is shown as follows:

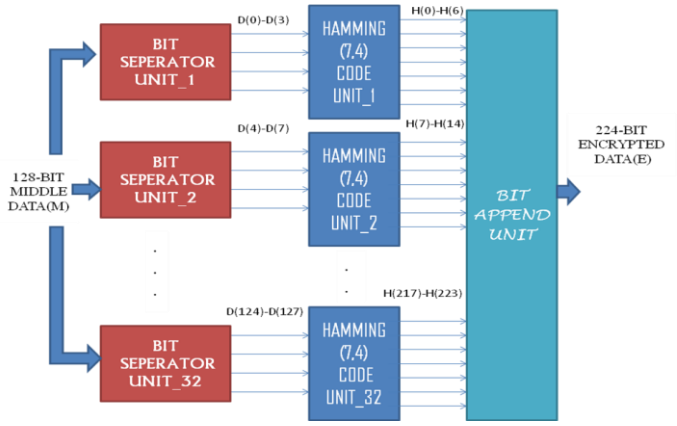

Fig-6: Block diagram Hamming (224,128) code Encryption Unit

Flow chart for modified iterated product cipher unit:

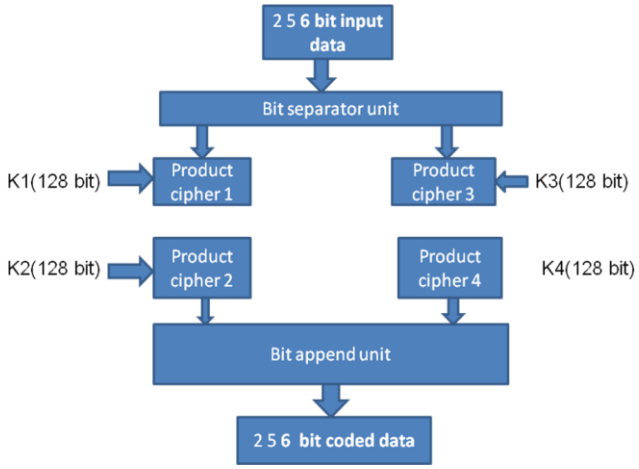

Fig-7 Flow chart of the modified iterated product cipher

Each product cipher unit is having the following flow chart:

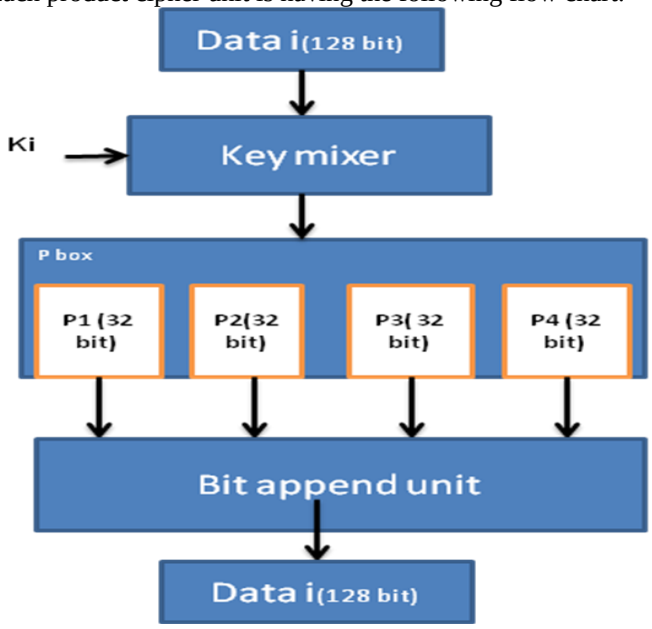

Fig-8 Flow chart of the product cipher unit

Where i varies from 1 to 4.

2. RESULTS AND DISCUSSION

The VHDL code of the proposed project is compiled, synthesized and simulated using Xilinx ISE 9.2i software and the desired results have been obtained.

The simulation result of the 128-bit digital data generation unit is given as follows:

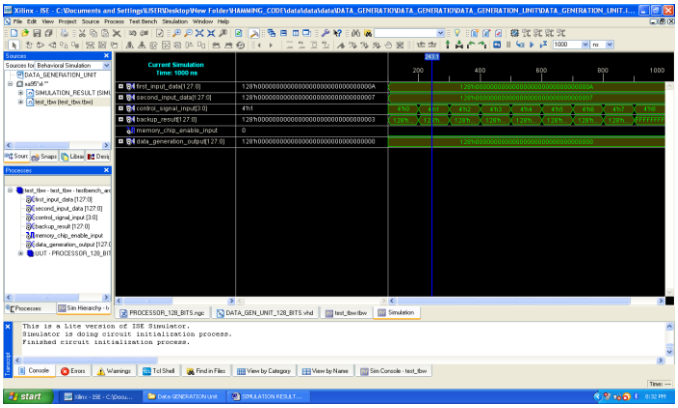

Fig-9: Simulation result of data generation unit for C='0'

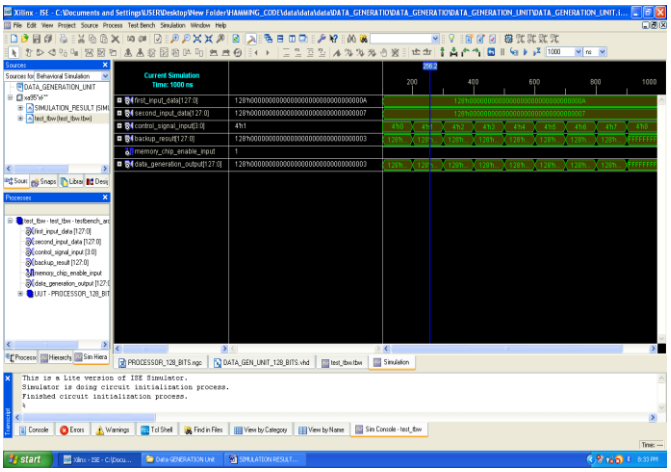

Fig-10: Simulation result of data generation unit for C='1'

The simulation result of the 128-bit digital data given to the DES encryption block to produce 128-bit middle data is shown as follows:

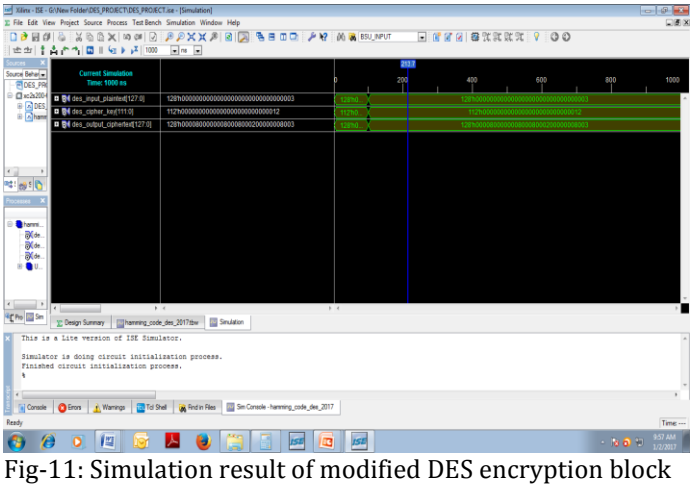

Fig-11: Simulation result of modified DES encryption block

The simulation result of the 128-bit middle data given to the Hamming (224,128) code encryption block to produce 224-bit encrypted data is shown as follows:

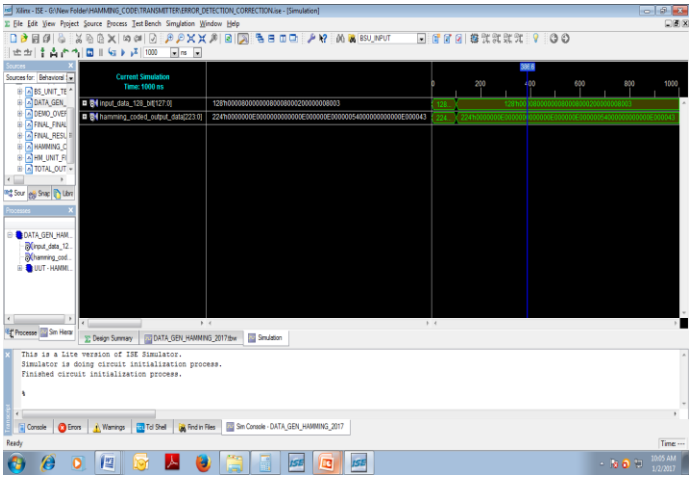

Fig-12: Simulation result of modified Hamming (224,128) code encryption block

The simulation result of the 224-bit to 256-bit converter is given as follows:

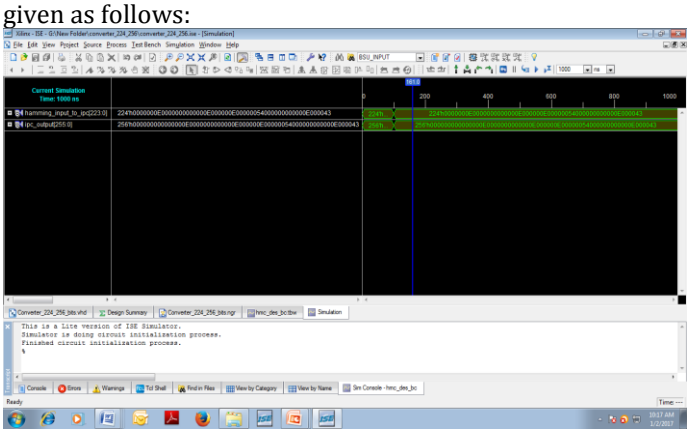

Fig-13: Simulation result of 224-bit to 256-bit converter

The simulation result of modified iterated product cipher unit is given as follows:

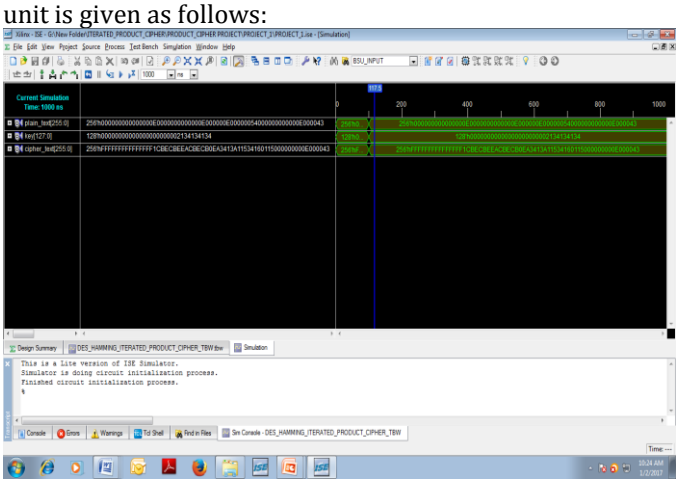

Fig-14: Simulation result modified iterated product cipher

The comparison study has been done based on the maximum combinational path delays of different data security algorithms obtained from the Xilinx software written VHDL code which is shown as follows:

Table -1: Comparison study

Name of the data security algorithm	Maximum combinational path delay found from the latest work (in ns)-T1	Maximum combinational path delay obtained from the proposed work (in ns)-T2	Complexity in terms of threshold value of Maximum combinational path delay	Security Level on the basic of Complexity
SUBSTITUTION CIPHER	1.5	----------	Low	Low
TRANSPOSITION CIPHER	5.4	----------	Low	Low
HAMMING CODE	7.4	8.468	----	----
MODIFIED DES	10.3	13.121	----	----
ITERATED PRODUCT CIPHER	----------	8.4	----	----
PROPOSED ALGORITHM	----------	29.989	Very High	Very High

3. CONCLUSIONS

The data generation unit is able to generate the 128-bit digital data as per the requirement and it is fed to the data security unit for its encryption. As the proposed design is having the combined effect of modified DES,Hamming (224,128) code and modified iterated product cipher data security techniques, the security level is very high as compared to the design having individual data security technique. Due to the use of 128-bit key size in modified iterated product cipher and the enhancement of key size in modified DES from 56-bits to 112-bits, the design is having more immunity towards the Brute-Force Attack.

REFERENCES

[1] W.Stallings,"Cryptography and Network Security", 2nd Edition, Prentice Hall.

[2] Christof Paar, Jan Pelzl, "The Data Encryption Standard (DES) and Alternatives", "Understanding Cryptography", Springer.

[3] Bruce Schneir: Applied Cryptography, 2nd edition, John Wiley & Sons.

[4] A.Litwin, "Cryptography and Network Security" LOS Alamitos,CA:IEEE computer society press.

[5] Douglas L. Perry. "VHDL Programming by Examples", TMH.

[6] Hamacher, Vranesic, and Zaky. Computer Organization, 5th edition, New York: McGraw-Hill Companies.

[7] Soufiane Oukili,Seddik Bri,"FPGA implementation of Data Encryption Standard using time variable permutations", International Conference on Microelectronics (ICM),IEEE,pp.126-129,2015.

[8] J. G. Pandey,Aanchal Gurawa, Heena Nehra,A. Karmakar , "An efficient VLSI architecture for data encryption standard and its FPGA implementation",VLSI SATA,IEEE International Conference,pp.1-5,2016.

[9] Ramadhan J. Mstafa; Khaled M. Elleithy, "A highly secure video steganography using Hamming code (7, 4)", Systems, Applications and Technology Conference (LISAT), IEEE Conference,pp.1-6,2014.

[10] B.A. Farouzan, "Cryptography and Network Security", Tata McGraw Hill Publication.

[11] W.Diffie; M.E.Hell man, " New Directions in Cryptography" ,IEEE transaction theory, Nov, pp 644-654.

[12] RanjanBose, "Information Theory, Coding and Cryptography" , chapter-8 .

[13] Ke Wang ," An encrypt and decrypt algorithm implementation on FPGAS" ,IEEE, Department of information engineering, 2009.

[14] Garfinkel, S.L, "Public Key Cryptography" , Computer, IEEE, Volume: 29, Issue:6.

[15] H. Lee Kwang , " Basic Encryption and Decryption" , Computer and Electrical.